ASG
Software Solutions

ASG-DataManager™
Total Interface

Version 2.5
Publication Number: DMR0200-25-TOT
Publication Date: June 1981

ASG Worldwide Headquarters Naples Florida USA | asg.com | info@asg.com
1333 Third Avenue South, Naples, Florida 34102 USA Tel: 239.435.2200 Fax: 239.263.3692 Toll Free: 800.932.5536 (USA only)

ASG Documentation/Product Enhancement Fax Form

Please FAX comments regarding ASG products and/or documentation to (239) 263-3692.

Company Name	Telephone Number	Site ID	Contact name

Product Name/Publication	Version #	Publication Date
Product:		
Publication:		
Tape VOLSER:		

Enhancement Request:

ASG Support Numbers

ASG provides support throughout the world to resolve questions or problems regarding installation, operation, or use of our products. We provide all levels of support during normal business hours and emergency support during non-business hours. To expedite response time, please follow these procedures.

Please have this information ready:

- Product name, version number, and release number
- List of any fixes currently applied
- Any alphanumeric error codes or messages written precisely or displayed
- A description of the specific steps that immediately preceded the problem
- The severity code (ASG Support uses an escalated severity system to prioritize service to our clients. The severity codes and their meanings are listed below.)
- Verify whether you received an ASG Service Pack for this product. It may include information to help you resolve questions regarding installation of this ASG product. The Service Pack instructions are in a text file on the distribution media included with the Service Pack.

If You Receive a Voice Mail Message:

1. Follow the instructions to report a production-down or critical problem.
2. Leave a detailed message including your name and phone number. A Support representative will be paged and will return your call as soon as possible.
3. Please have the information described above ready for when you are contacted by the Support representative.

Severity Codes and Expected Support Response Times

Severity	Meaning	Expected Support Response Time
1	Production down, critical situation	Within 30 minutes
2	Major component of product disabled	Within 2 hours
3	Problem with the product, but customer has work-around solution	Within 4 hours
4	"How-to" questions and enhancement requests	Within 4 hours

ASG provides software products that run in a number of third-party vendor environments. Support for all non-ASG products is the responsibility of the respective vendor. In the event a vendor discontinues support for a hardware and/or software product, ASG cannot be held responsible for problems arising from the use of that unsupported version.

Business Hours Support

Your Location	Phone	Fax	E-mail
United States and Canada	800.354.3578	239.263.2883	support@asg.com
Australia	61.2.9460.0411	61.2.9460.0280	support.au@asg.com
England	44.1727.736305	44.1727.812018	support.uk@asg.com
France	33.141.028590	33.141.028589	support.fr@asg.com
Germany	49.89.45716.222	49.89.45716.400	support.de@asg.com
Singapore	65.6332.2922	65.6337.7228	support.sg@asg.com
All other countries:	1.239.435.2200		support@asg.com

Non-Business Hours - Emergency Support

Your Location	Phone	Your Location	Phone
United States and Canada	800.354.3578		
Asia	65.6332.2922	**Japan/Telecom**	0041.800.9932.5536
Australia	0011.800.9932.5536	**Netherlands**	00.800.3354.3578
Denmark	00.800.9932.5536	**New Zealand**	00.800.9932.5536
France	00.800.3354.3578	**Singapore**	001.800.3354.3578
Germany	00.800.3354.3578	**South Korea**	001.800.9932.5536
Hong Kong	001.800.9932.5536	**Sweden/Telia**	009.800.9932.5536
Ireland	00.800.9932.5536	**Switzerland**	00.800.9932.5536
Israel/Bezeq	014.800.9932.5536	**Thailand**	001.800.9932.5536
Japan/IDC	0061.800.9932.5536	**United Kingdom**	00.800.9932.5536
		All other countries	1.239.435.2200

ASG Web Site

Visit http://www.asg.com, ASG’s World Wide Web site.

Submit all product and documentation suggestions to ASG’s product management team at http://www.asg.com/asp/emailproductsuggestions.asp.

If you do not have access to the web, FAX your suggestions to product management at (239) 263-3692. Please include your name, company, work phone, e-mail ID, and the name of the ASG product you are using. For documentation suggestions include the publication number located on the publication’s front cover.

THE MANAGER FAMILY

MANAGER FAMILY – MSP brings you a unique offering – the MANAGER Family of dictionary driven software products. MANAGER Products and their dictionaries may be integrated to maximize the return on your software and on your dictionary investment.

USER CONFIGURED – Every MANAGER Product consists of a nucleus with optional additional facilities. You select the options you need, and MSP supplies your chosen configuration.

WORLD-WIDE SUPPORT – All MANAGER Products are backed by comprehensive support, documentation, education and user group activities on a world-wide basis.

CONTROLMANAGER®

The dictionary driven End User Facility for the MANAGER Family of products

DATAMANAGER®

The dictionary driven Data and Information Resource Management system, winner of the ICP $50 Million Award

DESIGNMANAGER®

The dictionary driven Interactive Data and Enterprise Modeling system for logical and physical database design

METHODMANAGER™

The dictionary driven Information Engineering Methodology which automatically navigates you and your users through the systems life cycle

PROJECTMANAGER®

The dictionary driven Project Resource Management and Budgetary Control system

SOURCEMANAGER™

The dictionary driven On-line Application Development system

TESTMANAGER®

The Program/Structure Testing and Test Data Generation system

The MSP MANAGER FAMILY – designed for the future; implemented today.

MSP TRADEMARKS

The following are trademarks of Management Systems and Programming Ltd:

- CONTROLMANAGER (Registered as a United States Trademark at the U.S. Patent and Trademark Office.)
- DATAMANAGER® (Registered as a United States Trademark at the U.S. Patent and Trademark Office. Registered in Canada. Registered in the United Kingdom.)
- DESIGNMANAGER (Registered as a United States Trademark at the U.S. Patent and Trademark Office.)
- METHODMANAGER
- PROJECTMANAGER® (Registered as a United States Trademark at the U.S. Patent and Trademark Office. Registered in Canada.)
- SOURCEMANAGER
- TESTMANAGER® (Registered as a United States Trademark at the U.S. Patent and Trademark Office. Registered in Canada.)
- TEXTMANAGER
- InfoBank
- InfoDictionary
- InfoSystem
- InfoView.

TRADEMARK ACKNOWLEDGEMENTS

The following trademarks may appear in any one or more MSP Publications:

- IBM is a Registered Trademark of International Business Machines Corporation
- The LIBRARIAN is a Registered Trademark of Applied Data Research Inc
- PANVALET is a Trademark of Pansophic Systems Inc
- SYSTEM 2000 is a Registered Trademark of SAS Institute Inc.

PREFACE

This manual is one of a series describing DATAMANAGER, one of the MANAGER Family of dictionary-driven products developed by MSP for use on IBM System/360, System/370, 30xx and 4300 series, and plug compatible, machines. The manual describes the TOTAL Interface facility; a facility (additional to the basic set-up, maintenance and interrogation features) which enables the user to include TOTAL database and data set data definitions in the dictionary, and to produce TOTAL source language data descriptions directly from DATAMANAGER data definitions.

This edition supersedes the second edition of the manual together with its Amendment Lists 1 to 4 and all TOTAL Interface User Notices issued during the currency of that edition. This manual relates to DATAMANAGER Release 5.0.0 and CONTROLMANAGER Release 1.0.0.

MSP provides Maintenance Service for MANAGER Products in IBM OS and DOS and Siemens BS2000 environments, where the release of OS or DOS or BS2000 in use is defined in Appendix 2 of the MANAGER Products Installation in OS Environments, Installation in DOS Environments or Installation in BS2000 Environments manual in the table column headed "Compatible Release Level". MSP's Maintenance Service for a particular OS, DOS or BS2000 environment (Compatible Release Level) will continue for a period equal to or greater than the IBM or Siemens support for that environment. Particular MANAGER Products selectable units interface with certain IBM and/or Siemens software products that run within OS and/or DOS and/or BS2000 environments and/or with other vendor software: the Compatible Release Levels at which these products interface with MANAGER Products and the support provided by MSP are also documented in Appendix 2 to the MANAGER Products Installation in OS Environments, Installation in DOS Environments and Installation in BS2000 Environments manuals. Throughout MSP's technical documentation, the terms OS and DOS respectively cover all those variants of OS and of DOS for which MSP has defined a Compatible Release Level.

The terminals currently supported by MANAGER Products are defined in Appendix 2 of the Installation manuals listed above.

It is assumed that the reader has a knowledge of DATAMANAGER to the extent covered by the User's Guide, and is familiar with TOTAL/8.

Chapter 1 of this manual summarizes the interfaces between DATAMANAGER and TOTAL.

Chapter 2 discusses very briefly the concept of TOTAL databases and illustrates how a TOTAL database can be defined to DATAMANAGER.

Chapter 3 gives the specifications of the DATAMANAGER data definition statements for TOTAL databases and data sets.

Chapter 4 describes the interface between TOTAL and the DATAMANAGER Source Language Generation facility, which enables TOTAL DBGEN source statements to be generated from the data dictionary.

Chapter 5 specifies the direct relationships that exist between TOTAL and DATAMANAGER data definitions.

Appendix 1 describes an installation macro that permits the TOTAL Interface to be tailored to an installation's own standards.

The notation used in the specification of DATAMANAGER commands is described on page viii.

To assist you to make full use of this manual, the Contents table following this Preface is supported by a combined keyword index and usage index under the heading 'Usage Directory' at the back of the manual. The Usage Directory provides a means of accessing information by word occurrence or by function.

For the storage and job control requirements for installing and running DATAMANAGER with the TOTAL Interface facility, reference should be made to the Installation in OS Environments manual or the Installation in DOS Environments manual, as appropriate.

A range of manuals is available covering the MANAGER Family of Products. Details of the manuals and other documentation available are published every six months (at the end of June and the end of December) in the MSP Documentation Bulletin, which is distributed to all Users.

CONTENTS

NOTATION FOR STATEMENT FORMATS

In all MANAGER Products manuals, the following notation is used in the specification of statement formats (for commands and member definition statements):

- all words printed in capitals are statement identifiers or keywords that must be present in full or truncated form in the circumstances stated in the statement specification. The extent beyond which a word must not be truncated (because it would become ambiguous) is indicated by underlining of the characters that must be retained. (But note that truncation limits shown are to avoid ambiguity within the Product concerned and between that Product and CONTROLMANAGER; they could be affected by the presence of other MANAGER Products, and particularly by the presence of User Defined Commands or User Defined Syntax.)

- all words printed in lower case are variables for which the user must substitute a value consistent with the specification

- material enclosed in square brackets [ ] is an option which may be included or omitted as required

- braces { } indicate that a choice must be made of one of the options enclosed within them

- three periods or full stops ... indicates that the material they immediately follow may be repeated. Where ... immediately follows a closing square bracket or brace, the material that can be repeated is bounded by that square bracket or brace and the corresponding opening square bracket or brace. If material can be repeated only a limited number of times the repetition permitted is stated in the specification.

- other punctuation marks and symbols must be coded as shown, subject to the implications of any square brackets or braces enclosing them; except that where a single quote, ', is shown, a double quote, ", can alternatively be used, provided that the opening and closing quotes of any pair of quotes are the same character (single quote or double quote). Quote characters are used in syntax as delimiters, marking the beginning and end of a string of characters. Your Systems Administrator may have specified an additional character that may be used as an alternative delimiter character. You can find out if this has been done by entering the command:

  ```
  QUERY STRING-DELIMITER
  ```

CHAPTER I DATAMANAGER/TOTAL INTERFACE FACILITIES

DATAMANAGER's TOTAL Interface provides the following facilities for users in a TOTAL environment:

- the ability to define TOTAL/8 databases to DATAMANAGER, to hold the definitions in the data dictionary and to process them by the standard DATAMANAGER commands

- the ability to generate from the data dictionary, and to insert into the required source library, either complete TOTAL/8 database source statements or individual TOTAL data set descriptions, ready for processing by the TOTAL DBGEN program.

The ability to define a TOTAL database demands a further member type in DATAMANAGER. This member type is TOTAL-DATABASE, which in the member type hierarchy comes between PROGRAM or MODULE and FILE. Also required are variations of the FILE data definition statement, to cater for TOTAL data sets. A TOTAL data set can be defined as a FILE TOTAL MASTER or as a FILE TOTAL VARIABLE. The TOTAL-DATABASE data definition statement and the relevant formats of the FILE data definition statement are further discussed in Chapter 2 and specified in Chapter 3.

So that the definitions of TOTAL databases can be processed by DATAMANAGER in the same way as other members of the data dictionary, the keywords:

- TOTAL-DATABASES
- TOTAL-MASTER-FILES
- TOTAL-VARIABLE-FILES
- TOTAL-FILES

are added to member-type keywords available for use in the following basic DATAMANAGER commands:

- BULK
- GLOSSARY
- LIST
- PERFORM
- REPORT
- WHICH.

In addition, the keywords LINK and CONTROL are available for use in the VIA clause of the WHICH and WHAT commands. An interrogation that includes VIA LINK refers to either of the following clauses:

- the LINKED-TO clause of the TOTAL master file data definition statement
- the LINK clause of the TOTAL variable file data definition statement.

The LINK and CONTROL keywords are additional to those described in the specifications of the WHICH and WHAT commands in section 3.2 of the DATAMANAGER User's Guide, and they are specifically for use with the TOTAL Interface.

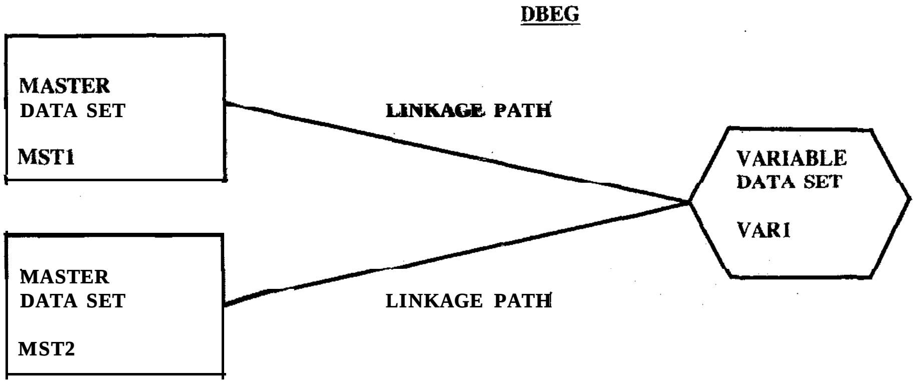
The ability to generate TOTAL source statements from data dictionary members requires the use of the Source Language Generation facility, which is described in a separate manual. The Source Language Generation manual describes the basic version of the facility, which can output data descriptions in COBOL, PL/I or Assembler. The enhancements to the facility to enable it to output TOTAL data descriptions are discussed in Chapter 4 of this manual.

CHAPTER 2 TOTAL DATABASES AND DATAMANAGER

The'ability to give a complete definition of a TOTAL/4, TOTAL/7 or TOTAL/8 database is available within DATAMANAGER.

The TOTAL database system is basically a system of linked files, or data sets, grouped together and referred to as a database. A TOTAL database consists of one or more master data sets with any number of variable data sets. The variable data sets are accessed through the master data sets via a system of linking.

For example, a simple database called DBEG containing two master data sets and one variable data set can be represented by the following schematic:



For DATAMANAGER the above structure would be represented as a TOTAL-DATABASE member containing three FILE members. The separate FILE members would contain details of the data set types and linkages with the other data sets. The records of the data sets and the separate elements making' up the records would be defined using the basic DATAMANAGER GROUP and ITEM data definition statements (these are not shown in the example).

If certain assumptions are made as to the characteristics and interrelationships of the data sets, the data definition statements could be as follows:

For DBEG:

```
TOTAL-DATABASE
CONTAINS MST1, MST2, VAR1
SHARE-IOAREA 'MI01', 'VI01'
```

For MST1:

```
FILE
TOTAL     MASTER
CONTAINS MST1DATA
CONTROL MST1CTRL
LINKED-TO   VAR1   LINK-NAME     'VI'
DEVICE    2314
TOTAL-LOGICAL-RECORDS     51273
LOGICAL-RECORD-LENGTH    230
LOGICAL-RECORDS-PER-BLOCK9
LOGICAL-BLOCKS-PER-TRACK3
RELATIVE-STARTING-TRACK     20
TOTAL-TRACKS  400
VTOC-TRACKS     20
IOAREA  'MI01'
```

For MST2:

```
FILE
TOTAL     MASTER
CONTAINS   MST2DATA
CONTROL MST2CTRL
LINKED-TO    VAR1     LINK-NAME    'V2'
DEVICE2314
TOTAL-LOGICAL-RECORDS115128
LOGICAL-RECORD-LENGTH95
LOGICAL-RECORDS-PER-BLOCK24
LOGICAL-BLOCKS-PER-TRACK3
RELATIVE-STARTING-TRACK     0220
TOTAL-TRACKS1600
VTOC-TRACKS     20
IOAREA   'MIO1'
;
```

For VAR1:

```
FILE
TOTAL      VARIABLE
CONTAINS    VAR1CODE,
            VAR1BASE,
            VAR1DATA
            ELSE  VAR1RCAA  IF  VAR1CODE  =  'AA'
            ELSE  VAR1RCBB  IF  VAR1CODE  =  'BB'
LINK   VAR1LNK1  TO  MST1    LINK-NAME   'V1',
       VAR1LNK2  TO  MST2    LINK-NAME   'V2'
DEVICE2314
TOTAL-LOGICAL-RECORDS1332
LOGICAL-RECORD-LENGTH29
LOGICAL-RECORDS-PER-BLOCK37
LOGICAL-BLOCKS-PER-TRACK7
VTOC-TRACKS  20
CYLINDER-LOAD-LIMIT80
IOAREA 'VIO1'
,
```

Such data definitions can be inserted into the data dictionary's source data set by INSERT or ADD commands, in the same way as any other DATAMANAGER data definitions, or by using CONTROLMANAGER's full screen editing capabilities.

The full specifications of the TOTAL-DATABASE data definition statement and of the variants of the FILE data definition statement that are applicable to TOTAL data sets are given in the next chapter.

CHAPTER 3 DATAMANAGER DATA DEFINITION STATEMENTS FOR TOTAL DATABASES

3.1 INTRODUCTION

The logical components of a TOTAL database can be defined in a DATAMANAGER data dictionary by the corresponding DATAMANAGER member types listed below:

TOTAL logical component	DATAMANAGER member type
Database	TOTAL-DATABASE
Master data set	FILE TOTAL MASTER
Variable data set	FILE TOTAL VARIABLE
Data record	GROUP
Data element	GROUP or ITEM*
Data item	ITEM

The specifications of the TOTAL-DATABASE, FILE TOTAL MASTER and FILE TOTAL VARIABLE data definition statements are given in sections 3.2 to 3.4 respectively. The DATAMANAGER GROUP and ITEM data definition statement specifications are given in the User's Guide, Chapter 4.

*Depending on the absence or presence of the keyword NO-GROUPS in the PRODUCE command: see Chapter 4.

3.2 THE TOTAL-DATABASE DATA DEFINITION STATEMENT

Format

```
TOTAL-DATABASE CONTAINS file-name [,file-name]...

[SHARE-IOAREA 'name' Cbl [,'name'[b]]...]

[TRACK-HOLD]

[ (LOGDD 'name'                                                        )1
  {LOGSIZE m                                                           }
  (LOGGING {FILE name SIZE m [DEVICE device1 [SYSNO nl}                )
           (NONE                                      )

[CTLX  (NONE                        )1
       {YES                         }
       (DEVICE  device  [SYSNO nl  )

[JCL {YES} ]
     {NO }

[common clauses1

{;}
{.}
```

where

file-name is the name of a FILE **member that defines a data set within the TOTAL database**

name **is a four-character name complying with the rules stated** in the TOTAL **manual**

b **is an unsigned integer, being the number 'of input/output buffers required for the** SHARE-IOAREA **'name'**

m is **an unsigned integer, being the size in bytes of the log buffer**

'device is one of the following:

2311
2314
3330
3340
3344
TAPE (valid on LOGGING clause only)

3350
3370
3375
3380
FBA

n **is an unsigned integer in the range 0 to 255**

common clauses are any of the following clauses, as defined in section 4.3 of the DATAMANAGER User's **Guide, in any order:**

- **-ACCESS-AUTHORITY**
- **-ADMINISTRATIVE-DATA**
- **- ALIAS**
- **- CATALOGUE**
- **- COMMENT**
- -DESCRIPTION
- -EFFECTIVE-DATE
- **- FREQUENCY**
- **- NOTE**
- **-OBSOLETE-DATE**
- **- QUERY**
- - SECURITY-CLASSIFICATION
- **- SEE**

Remarks

1 File-names must conform to the rules for member names stated in Chapter 2 of the DATAMANAGER User's Guide.

2 The CONTAINS keyword is followed by a list of one or more file-names. If two or more file-names are listed, each except the first in the list must be preceded by a comma. Entries in the list can additionally be separated by spaces.

3 The SHARE-IOAREA clause has the same meaning as the SHARE-IO option in TOTAL. The keyword is followed by a list of one or more four-character names, each of which can optionally be followed by an unsigned integer specifying the number of input/output buffers required for that named area. If two or more names are listed, each except the first in the list must be preceded by a comma. Entries in the list can additionally be separated by spaces.

4 If the Source Language Generation facility's PRODUCE command is used to generate TOTAL DBGEN source statements from this data definition, then for each SHARE-IOAREA named for which the number of input/output buffers is not specified, single buffering is assumed.

5 Validation checks are performed on SHARE-IOAREA names and on log file ~~names by DATAMANAGER's ENCODE command to ensure that the names~~ are four characters long and contain only characters valid for TOTAL (A to Z, 0 to 9, #, @, and currency symbol with internal code hexadecimal 5B). For SHARE-IOAREA names an additional check is made to ensure that the first three characters are not PRI.

6 The optional keyword TRACK-HOLD, the optional log clauses, the CTLX clause, and the JCL clause have the same meaning as the equivalent TOTAL Prologue Statements. No checks are performed by DATAMANAGER for consistent usage of TOTAL statements when the data definition is encoded. DATAMANAGER does not supply any default value if these clauses are omitted.

7 The LOGDD clause is valid in a multi-task environment only.

8 The LOGSIZE clause is valid in a single-task environment only. LOGSIZE 0 can be declared if the TOTAL logging feature is not in use.

9 The LOGGING clause DEVICE and SYSNO subordinate clauses have meaning only if TOTAL/8 is operating under DOS.

10 Common clauses, listed under Format above, can be present in any type of data definition statement; they are therefore defined separately, in section 4.3 of the DATAMANAGER User's Guide. Not more than one of each of these clauses can be declared. If a common clause has a subordinate clause or keyword, the subordinate clause identifier or subordinate keyword must not be truncated to an extent where it becomes ambiguous with any other clause identifier or other keyword available in the data definition syntax for this member type.

11 The clauses of the TOTAL-DATABASE data definition statement can be declared in any order between the statement identifier and the terminator.

12 A record containing the database's data definition statement can be inserted into the data dictionary's source data set by a suitable command (see DATAMANAGER User's Guide, Chapter 3) and an encoded record can subsequently be generated and inserted into the data entries data set. If, when the encoded record is generated, any member whose name appears in the database's data definition statement has no data entries record, DATAMANAGER creates a dummy data entries record for that member. Unless the member's name appears in a SEE clause, the dummy record is created as a dummy FILE TOTAL MASTER. If the member's name appears in a SEE clause, the dummy record is created as a dummy ITEM.

Example

An example is given in Chapter 2.

3.3 THE FILE DATA DEFINITION STATEMENT FOR TOTAL MASTER DATA SETS

Format

```
FILE TOTAL MASTER
CONTAINS  member-name  [KNOWN-AS  Local-name1
          C,member-name  [KNOWN-AS  Local-name]]...
[ CONTROL member-name]
[CONTROL-INTERVAL   nl   [CONTROL-INTERVAL-SIZE    nl
[ACCESS-METHOD {BDAM}1
               {VSAM}
[LINKED-TO f i Le-name [FOLLOWING {KEY } 1 LINK-NAME 'xx"
                                  {DATA}

   [,file-name [FOLLOWING          } LINK-NAME 'xx']...
                          & : A    }

   [FOLLOWING {KEY }11
              {DATA}

[DEVICE device1
[TOTAL-LOGICAL-RECORDS n] C-LENGTH nl
[LOGICAL-RECORDS-PER-BLOCK  nl [LOGICAL-BLOCKS-PER-TRACK  n]
[DISK-EXTENTS  ml  [RELATIVE-STARTING-TRACK  nl
[VTOC-TRACKS n l [CYLINDER-LOAD-LIMIT n l
[OLD-FILE] [TOTAL-TRACKS n]
[IOAREA 'xxxx']
[common clauses1
{;}
{.}
```

where

local-name is a name, conforming to the rules for member names stated in section 2.4 of the DATAMANAGER User's Guide, that is to be used instead of the name or alias of the contained member, when source language data descriptions are generated from this data definition by the DATAMANAGER Source Language Generation facility. The local-name is not separately recorded in the data dictionary (that is, no dummy data entries record and no index record is created for local-name when the data definition in which it appears is encoded) so local-name cannot be interrogated and can be the same as another name, an alias or a catalog classification in the data dictionary. The local-name is the name by which the contained member is known only within the file defined by this data definition.

member-name is the name of a data dictionary GROUP or ITEM member

n is an unsigned integer that can be derived by reference to the TOTAL manual

file-name is the name of a data dictionary FILE member that defines a TOTAL VARIABLE file to which this file is linked

xx is a two-character identifier

device is one of the following:

2311
2314
3330
3340
3344

3350
3370
3375
3380
FBA

m is an unsigned integer in the range 1 to 16

xxxx is a four-character name that appears in the SHARE-IOAREA clause of the TOTAL-DATABASE data definition from which this file is referenced

common clauses are any of the following clauses, as defined in section 4.3 of the DATAMANAGER User's Guide, in any order:

- ACCESS-AUTHORITY
- ADMINISTRATIVE-DATA
- ALIAS
- CATALOGUE
- COMMENT
- DESCRIPTION
- EFFECTIVE-DATE
- FREQUENCY
- NOTE
- OBSOLETE-DATE
- QUERY
- SECURITY-CLASSIFICATION
- SEE

Remarks

1 Field-names, group-names and item-names must conform to the rules for member names stated in Chapter 2 of the DATAMANAGER User's Guide.

2 The keywords TOTAL MASTER must immediately follow the FILE statement identifier; otherwise, the syntax cannot be accurately checked.

3 The clauses declared following the TOTAL MASTER keywords can appear in any order.

4 The CONTAINS clause lists the names of the members that constitute this file's records. If two or more member-names are listed, each except the last in the list must be followed by a comma. Entries in the list can additionally be separated by spaces.

5 The member-name in the CONTROL clause is the name of the data dictionary group or item member that defines the control field. It must be referenced directly or indirectly from the CONTAINS clause. For TOTAL/4 the control field must be the first group or item within the file's records.

6 If the Source Language Generation facility's PRODUCE command is used to generate DBGEN source statements, and the CONTROL clause is omitted from the data definition, then the first member named in the CONTAINS clause is taken to be the control field; unless the PRODUCE command contains the keyword NO-GROUPS, in which case:

- if the first member named in the CONTAINS clause is an ITEM, that member is taken to be the control field
- if the first member named in the CONTAINS clause is a GROUP, the first item directly or indirectly contained by that group is taken to be the control field.

If TOTAL/4 DBGEN source statements are being generated and the CONTROL clause specifies a member other than that generated as above as the control field, the warning messages:

```
member-name SPECIFIED CONTROL FIELD NOT FIRST ELEMENT
member-name ASSUMED TO BE CONTROL FIELD - NAME HAS BEEN
            CHANGED
```

are output.

7 The CONTROL-INTERVAL, CONTROL-INTERVAL-SIZE and ACCESS-METHOD clauses have the same meaning as in TOTAL/S.

8 The LINKED-TO clause specifies the linkages between this master file and variable files, If the Source Language Generation facility's PRODUCE command is used to generate TOTAL DBGEN source statements from this data definition, a check is made that file-name is a variable file. The link-name xx appears also in file-name's linkage specification.

9 The LINKED-TO keyword is followed by a list of one or more linkage specifications. If two or more linkage specifications are listed, each except the last in the list must be followed by a comma. Entries in the list can additionally be separated by spaces.

10 The FOLLOWING subordinate clause of the LINKED-TO clause specifies the position of the link path field. FOLLOWING KEY specifies that the link path immediately follows its associated key field. FOLLOWING DATA specifies that the link path field immediately follows the last data element in that portion of the record with which it is associated. If the clause is omitted, DATAMANAGER assumes a default of FOLLOWING KEY.

11 A FOLLOWING clause that immediately follows file-name specifies the position of the link statements for that particular file-name only. The FOLLOWING **clause** that follows the last LINK-NAME clause applies to all links for this TOTAL MASTER file and thus overrides any preceding FOLLOWING clauses specified for individual file-names.

12 The optional clauses **and** keywords from DEVICE to IOAREA in the statement format have the same meanings as in TOTAL.

13 If the DEVICE clause is omitted. a default value of 2314 for device is assumed, unless otherwise tailored by the DEVICE keyword of the installation macro DGTOT (see Appendix 1).

14 The DISK-EXTENTS, RELATIVE-STARTING-TRACK and VTOC-TRACKS clauses are always allowed by DATAMANAGER but have meaning for TOTAL only if TOTAL is operating under DOS.

15 If the DISK-EXTENTS clause is specified with a value of m greater than 16. DATAMANAGER substitutes a value of 1 when the member is encoded,. and outputs the warning message:

```
integer OUT OF RANGE - 1 ASSUMED
```

16 The name xxxx in the IOAREA clause must not also appear in the IOAREA clause of a TOTAL VARIABLE file data definition relating to the same database. Validation of the name performed by DATAMANAGER is as **follows:**

- by the ENCODE command:

  - that the name is four characters, from the character set A to Z, 0 to 9, #, @ and currency symbol of internal code hexadecimal 5B
  - that the first three characters are not PRI

- by the PRODUCE command of the Source Language Generation facility:

  that the name does not appear in the IOAREA clause of any TOTAL VARIABLE file processed for the same database by the same PRODUCE command.

17 Except where otherwise stated. no default value is supplied by DATAMANAGER for any clause that is omitted from the data definition.

18 When TOTAL DBGEN source statements are generated by the PRODUCE command. DATAMANAGER generates a root field. The name of the root field is the generated file-name (see section 4.4) suffixed by ROOT.

19 Common clauses, listed under Format above. can be present in any type of data definition statement; they are therefore defined separately. in section 4.3 of the DATAMANAGER User's Guide. Not more than one of each of these clauses can be declared. If a common clause has a subordinate clause or keyword, the subordinate clause identifier or subordinate keyword must not be truncated to an extent where it becomes ambiguous with any other clause identifier or other keyword available in the data definition syntax for this member type.

20 A record containing the file's data definition statement can be inserted into the data dictionary's source data set by a suitable command (see DATAMANAGER User's Guide, Chapter 3) and an encoded record can subsequently be generated and inserted into the data entries data set. If, when the encoded record is generated, any member whose name appears in the tile's data definition has no data entries record, a dummy data entries record is created for that member. The dummy record is created as a dummy ITEM if its name appears in the CONTAINS clause and/or the CONTROL clause and/or the SEE clause, or as a dummy FILE TOTAL VARIABLE if its name appears in the LINKED-TO clause.

Examples

Examples are given in Chapter 2. In those examples, member-names in the CONTAINS clause include file-name prefixes. An implication of this is that, if the Source Language Generation facility's PRODUCE command is used to generate TOTAL source statements from these data definitions, then either the installation macro DGTOT should be tailored to include PREFIX=NO (see Appendix I) or the PRODUCE command should include the clause OMITTING FILE-NAME-PREFIX (see Chapter 4). Otherwise, on generation, the member-names would be shortened to four characters by the omission of middle characters, and the file-name prefix would be added to the resulting four character names; so that, for example, the name generated from the member-name MST1DATA would be MSTIMSTA.

3.4 THE FILE DATA DEFINITION STATEMENT FOR TOTAL VARIABLE DATA SETS

Format

```
FILE TOTAL VARIABLE
CONTAINS   member-name   [KNOWN-AS   local   name1
         C,member-name   [KNOWN-AS   Local-name]]...
         C,member-name   ELSE   member-name   IF   item-name   =   'aa'
         [ELSE   member-name   IF   item-name   = 'aa']...]
[CONTROL-INTERVAL   n]   [CONTROL-INTERVAL-SIZE   n]
[ACCESS-METHOD {BDAM}1
               {VSAM}

LINK   member-name-l   TO   file-name
    [FOLLOWING {KEY }1  LINK-NAME  'xx'
               {DATA}

    C,member-name-l   TO   file-name
    [FOLLOWING {KEY }1  LINK-NAME 'x']...
               {DATA}

    [FOLLOWING {KEY }1
               {DATA}

[DEVICE  device1
[TOTAL-LOGICAL-RECORDS nl [LOGICAL-RECORD-LENGTH nl
[LOGICAL-RECORDS-PER-BLOCK n l [LOGICAL-BLOCKS-PER-TRACK n l
[DISK-EXTENTS  ml  [RELATIVE-STARTING-TRACK  nl
[VTOC-TRACKS nl [CYLINDER-LOAD-LIMIT pl
[OLD-FILE]  [TOTAL-TRACKS  n]
[IOAREA 'xxxx']
[common  clauses1
{;}
{.}
```

where

local-name is a name, conforming to the rules for member names stated in section 2.4 of the DATAMANAGER User's Guide, that is to be used instead of the name or alias of the contained member, when source language data descriptions are generated from this data definition by the DATAMANAGER Source Language Generation facility. The local-name is not separately recorded in the data dictionary (that is, no dummy data entries record and no index record is created for local-name when the data definition in which it appears is encoded) so local-name cannot be interrogated and can be the same as another name, an alias or a catalog classification in the data dictionary. The local-name is the name by which the contained member is known only within the file defined by this data definition.

member-name is the name of a data dictionary GROUP or ITEM member that forms part of this tile's records

item-name is the name of a data dictionary ITEM member that is the first field within this file's records and is the TOTAL RECORD-CODE

aa is the two-character code identifying the record type

n is an unsigned integer that can be derived by reference to the TOTAL manual member-name-l is the name of a data dictionary GROUP or ITEM member that is declared in the CONTAINS clause or is referenced directly or indirectly from one of the members declared in the CONTAINS clause

file-name is the name of a data dictionary FILE member that defines a TOTAL MASTER file to which this file is linked

xx is a two-character identifier

device is one of the following:

2311
2314
3330
3340
3344

3350
3370
3375
3380
FBA

m is an unsigned integer in the range 1 to 16

p is an unsigned integer in the range 1 to 80

xxxx is a four-character name that appears in the SHARE-IOAREA clause of the TOTAL-DATABASE data definition from which this file is referenced

common clauses are any of the following clauses, as defined in section 4.3 of the DATAMANAGER User's Guide, in any order:

- ACCESS-AUTHORITY
- ADMINISTRATIVE-DATA
- ALIAS
- CATALOGUE
- COMMENT
- DESCRIPTION
- EFFECTIVE-DATE
- FREQUENCY
- NOTE
- OBSOLETE-DATE
- QUERY
- SECURITY-CLASSIFICATION
- SEE

Remarks

1 File-names, group-names and item-names must conform to the rules for member names stated in Chapter 2 of the DATAMANAGER User's Guide.

2 The keywords TOTAL VARIABLE must immediately follow the FILE statement identifier; otherwise, the syntax cannot be accurately checked.

3 The clauses declared following the TOTAL VARIABLE keywords can appear in any order; except that as the CONTAINS clause includes its associated ELSE and IF clauses, other clauses must not intervene.

4 For a TOTAL VARIABLE file with base data only, the CONTAINS clause contains a list of one or more member-names. If two or more member-names are listed, each except the first in the list must be preceded by a comma; the comma can optionally be preceded and/or followed by spaces.

5 For a TOTAL VARIABLE file with record-coded records, the CONTAINS keyword is followed by a list of two or more member-names, each except the last in the list being the name either of an item which is a TOTAL base-data element or of a group which (directly 'or indirectly) contains items which are TOTAL base-data elements. Each member-name except the first in the list must be preceded by a comma; the comma can optionally be preceded and/or followed by spaces.

The last member-name in the list is an alternative entry; that is, it has associated with it a number of ELSE and IF clauses, specifying alternative members (groups or items) and the conditions under which they are present in the file's records. These conditions are the characters contained in item-name, the two-character code identifying the record-type. Item-name must either be the same as the first member-name, or must be the name of the item directly or indirectly defined as the first within that member (if that member is a group). ELSE and IF clauses are not separated by commas. If the Source 'Language Generation facility's PRODUCE command is used to generate TOTAL DBGEN source statements from this data definition, this last element is generated with the length of the longest of the alternative members.

6 The CONTROL-INTERVAL, CONTROL-INTERVAL-SIZE and ACCESS-METHOD clauses have the same meaning as in TOTAL/8 and are not relevant to TOTAL/4 or TOTAL/7.

7 The LINK clause specifies the linkages between this variable file and master files. If the Source Language Generation facility's PRODUCE command is used to generate TOTAL DBGEN source statements from this data definition, a check is then made that file-name is a master file. The link-name xx appears also in file-name's linkage specification.

8 The LINK keyword is followed by a list of one or more linkage specifications. If two or more linkage specifications are listed, each except the last in the list must be followed by a comma. Entries'in the list can additionally be separated by spaces.

9 The FOLLOWING subordinate clause is part of the LINK clause. It specifies the position of the link path field. FOLLOWING KEY specifies that the link path field immediately follows its associated key field. FOLLOWING DATA specifies that the link path field immediately follows the last data element in that portion of the record with which it is associated. If the clause is omitted, DATAMANAGER assumes a default of FOLLOWING KEY.

10 A FOLLOWING clause that immediately follows file-name specifies the position of the link statements for that particular file-name only. The FOLLOWING **clause** that follows the last LINK-NAME **clause** applies to all links for this TOTAL VARIABLE file; it thus overrides any preceding FOLLOWING clauses specified for individual file-names.

11 The optional clauses and keywords from DEVICE to IOAREA in the statement format have the same meaning as in TOTAL.

12 If the DEVICE clause is omitted, a default value of 2314 is assumed, unless otherwise tailored by the DEVICE keyword of the installation macro DGTOT (see Appendix 1).

I3 The DISK-EXTENTS, RELATIVE-STARTING-TRACK and VTOC-TRACKS clauses are always allowed by DATAMANAGER but have meaning for TOTAL only if TOTAL is operating under DOS.

14 If the DISK-EXTENTS clause is specified with a value of m greater than 16, DATAMANAGER substitutes a value of 1 when the member is encoded, and outputs the warning message:

```
integer OUT OF RANGE - 1 ASSUMED
```

15 The name of xxxx in the IOAREA clause must not also appear in the IOAREA clause of a TOTAL MASTER file data definition relating to the same database. Validation of the name performed by DATAMANAGER is as follows:

- by the ENCODE command:

  - that the name is four characters, from the character set A to Z. 0 to 9, #, @ and currency symbol of internal code hexadecimal 5B.
  - that the first three characters are not PRI

- by the PRODUCE command of the Source Language Generation facility:

  - that the name does not appear in the IOAREA clause of any TOTAL MASTER file processed for the same database by the same PRODUCE command.

16 Except where otherwise stated, no default value is supplied by DATAMANAGER for any clause that is omitted from the data definition.

17 When TOTAL DBGEN source statements are generated by the PRODUCE command, DATAMANAGER generates a root field. The name of the root field is the file-name, shortened to four characters by dropping middle characters if necessary, suffixed by ROOT.

18 Common clauses, listed under Format above, can be present in any type of data definition statement; they are therefore defined separately, in section 4.3 of the DATAMANAGER User's Guide. Not more than one of each of these clauses can be declared. If a common clause has a subordinate clause or keyword, the subordinate clause identifier or subordinate keyword must not be truncated to an extent where it becomes ambiguous with any other clause identifier or other keyword available in the data definition syntax for this member type.

19 A record containing the file's data definition statement can be inserted into the data dictionary's source data set by a suitable command (see DATAMANAGER User's Guide, Chapter 3) and an encoded record can subsequently be generated and inserted into the data entries data set. If, when the encoded record is generated, any member whose name appears in the file's data definition has no data entries record, DATAMANAGER creates a dummy data entries record for that member. The dummy record is created as a dummy ITEM if its name appears in the SEE clause and/or in the CONTAINS clause and/or in the LINK clause immediately before a TO keyword or as a dummy FILE TOTAL MASTER if its name appears in the LINK clause immediately after a TO keyword.

Example

An example is given in Chapter 2. In that example, member-names in the CONTAINS and LINK clauses include file-name prefixes. An implication of this is that, if the Source Language Generation facility's PRODUCE command is used to generate TOTAL source statements from these data definitions, then either the installation macro DGTOT should be tailored to include PREFIX = NO (see Appendix 1) or the PRODUCE command should include the clause OMITTING FILE-NAME-PREFIX (see Chapter 4). Otherwise on generation, the member-names would be shortened to four characters, and the file-name prefix would be added to the resulting four character names; so that for example, the name generated from the member VARICODE would be VARIVADE.

CHAPTER 4 TOTAL SOURCE LANGUAGE GENERATION FROM DATAMANAGER

4.1 INTRODUCTION

DATAMANAGER can generate TOTAL source language statements from encoded data dictionary members for input to the TOTAL DBGEN program. This is achieved by use of the PRODUCE command, described in the separate DATAMANAGER manual entitled Source Language Generation.

The Source Language Generation manual describes the general form of the PRODUCE command, and gives the detailed specifications of the command for the generation of record layouts and of COBOL, PL/I and Assembler source language data descriptions. This chapter describes the variation of the command for the generation of TOTAL source language statements: it should be read in conjunction with the Source Language Generation manual (Chapter 2 and the relevant parts of Chapter 1).

An installation macro, DGTOT, permits the output from this variation of the PRODUCE command to be tailored to conform to the particular installation's standards. DGTOT is described in Appendix 1.

The specification of the PRODUCE command for the generation of TOTAL source language statements is given in section 4.2. Further details of TOTAL source language generation from TOTAL-DATABASE members, and from FILE TOTAL MASTER members and FILE TOTAL VARIABLE members, are given in sections 4.3 and 4.4 respectively. Section 4.5 discusses the generation of record layouts and programming source language data descriptions from TOTAL-DATABASE, FILE TOTAL MASTER and FILE TOTAL VARIABLE members.

4 . 2 SPECIFICATION OF THE PRODUCE COMMAND FO R TOTAL SOURCE LANGUAGE GENERATION

Purpose

To generate TOTAL/4, TOTAL/7 or TOTAL/8 source language statements for subsequent input to the TOTAL DBGEN program.

Format

```
PRODUCE ⎧TOTAL-4⎫ FROM member-name [AS library-name1
        ⎨TOTAL-7⎬
        ⎩TOTAL-8⎭

[,member-name[AS library-name1 ]...

[⎧NOGROUPS ⎫1
 ⎩NO-GROUPS⎭

Ccontrol-options1

⎧;⎫
⎩.⎭
```

where

member-name is the name of an encoded data dictionary member that is a TOTAL-DATABASE or a FILE TOTAL MASTER or a FILE TOTAL VARIABLE

library-name is the name to be given to the generated library member in the output data set. The name must not be more than eight characters. The first character must be alphabetic or one of the characters #, £ (or local currency symbol with the internal code hexadecimal 5B), % or @.

control-options is as defined in Chapter 2 of the Source Language Generation manual, with the addition of:

- the keyword

  FILE-NAME-PREFIX

  to the output-form-l and output-form-2 options of the GIVING and OMITTING clauses

- the clause

  ```
  LINKS FOLLOWING ⎧KEY ⎫
                  ⎩DATA⎭
  ```

  to the output-form-l options of the GIVING clause.

  The GIVING and OMITTING clauses are defined in section 2.2.3 of the Source Language Generation manual.

Remarks

1 The command identifier, PRODUCE, must be immediately followed by one of the context keywords TOTAL-4, TOTAL-7 or TOTAL-8, to indicate which version of the TOTAL language is to be generated. The context keyword must be immediately followed by the FROM clause (with its optional subordinate AS clauses, if applicable). All other clauses are optional and can be in any order.

2 Up to a maximum of 16 member names can be declared in the FROM clause. If two or more are declared, each except the first must be preceded by a comma; the comma can optionally be preceded and/or followed by spaces. Member names are processed in the order in which they appear in the FROM clause.

3 Acceptance of the PRODUCE command is in respect of each member-name individually. If member-name:

- is not encoded, or
- is not present in the data dictionary, or
- is of a member type that is not valid in the context, or
- is protected against access by the user (see remark 4)

a message is output, no generation takes place in respect of that member-name, and processing continues with the next member-name or command.

4 Acceptance of the PRODUCE command is subject to access security levels (for each member-name individually, as stated in remark 3). See the specification of the PROTECT command in the DATAMANAGER User's Guide, and the description of the security system in section 1.7 of that User's Guide. If a member-name has an access security level higher than the user's (general or specific) security level, the command is rejected in respect of that member-name. If the command can be accepted in respect of member-name, but a reference path from member-name includes a protected member with an access level higher than the user's security level, the reference path is followed to its end to determine the total storage space required, but the name of no member in that reference path beyond the last member to which the user has access is given; instead, a filler name is generated.

5 AS clauses are relevant only if TOTAL source language statements are being produced and written to an output data set. (Statements can be generated for listing on a printer or terminal only, without being written to an output data set: see the control-options specifications.)

6 Each AS clause present in the command relates only to the member-name that immediately precedes it. It declares a name under which the generated source language data description is to be catalogued in the output source library data set.

7 For each member-name for which no AS clause is specified, library-name is defaulted to member-name if member-name conforms to the length restrictions on library-name. The length restriction on library-name is a maximum of eight characters (unless tailored, see MEMLEN). If member-name is longer than the permitted maximum length for library-name, no generation takes place in respect of that member-name, a message is output, and processing continues with the next member-name or command.

8 Library-names, whether declared or defaulted, are not subject to any name editing, ALIAS or WITH-ALIAS clauses (see control-options) that may be present in the command.

9 The keyword NOGROUPS or NO-GROUPS specifies that TOTAL elements are to be generated only from ITEM members directly or indirectly contained by member-name. No TOTAL source language statements are generated for any GROUP members that are encountered in the reference paths from member-name to the ITEM members.

10 If the NOGROUPS/NO-GROUPS keyword is omitted, then:

- for TOTAL/4, TOTAL elements are generated for GROUP and ITEM members that are directly contained by a FILE member. Indirectly contained items are listed as comments. No TOTAL elements or comments are generated for indirectly contained groups.

- for TOTAL/7 and TOTAL/8, TOTAL elements are generated for all GROUP and ITEM members directly or indirectly contained by a FILE member, to give the complete data structure.

11 For TOTAL data sets, control fields are generated as stated in the FILE TOTAL MASTER data definition statement specifications in section 3.3.

12 GIVING FILE-NAME-PREFIX overrides the PREFIX = NO keyword usage of the DGTOT macro. (The supplied value of the PREFIX keyword is YES.) Generated data element names are prefixed with the four letters of the generated name of the containing file. For fuller details of the generation of data element names, see section 4.4.

13 OMITTING FILE-NAME-PREFIX overrides the PREFIX = YES keyword usage of the DGTOT macro. Generated data element names are not prefixed with the name of the containing file.

14 GIVING LINKS FOLLOWING KEY or GIVING LINKS FOLLOWING DATA overrides any contrary FOLLOWING clause in the data definition of any FILE member processed by the command. If the member-name in the command is a TOTAL-DATABASE, the GIVING LINKS clause is applied to all FILE members contained by that member.

15 Other control-options clauses are as stated in the Source Language Generation manual.

Examples

For examples of the PRODUCE command and the resulting output. see Chapter 4 of the DATAMANAGER Example Book.

4.3 GENERATION FROM TOTAL-DATABASE MEMBERS

TOTAL DBDL Prologue Statements are generated from TOTAL-DATABASE members named in the PRODUCE command.

The name generated for a database is the member name or alias (see section 2.2.3 of the Source Language Generation manual), subjected to any name editing specified in the PRODUCE command (see section 2.2.4 of the Source Language Generation manual), and then adjusted to six characters by removing the middle characters or by appending X characters.

Optional clauses (other than the informatory clauses) in the member's data definition result in the generation of the equivalent optional TOTAL statements. provided that:

- they are valid for the version of TOTAL being produced
- they are valid (for TOTAL) in the operating system environment (OS or DOS) under which the generation is taking place.

If a statement generated from an optional clause would be invalid for the version of TOTAL being produced, or invalid for the operating system environment, generation from that clause is suppressed.

If the optional SHARE-IOAREA clause is not completely specified, default values are applied as stated in the TOTAL-DATABASE data definition statement specifications in section 3.2.

DESCRIPTION clauses result in the generation of TOTAL source language comments, immediately following the Prologue Statements, if:

- the DESC keyword of the DGTOT macro has a value other than 0, and the PRODUCE command does not include OMITTING DESCRIPTIONS; or
- the PRODUCE command includes GIVING [s] DESCRIPTIONS (see section 2.2.3 of the Source Language Generation manual).

NOTE clauses result in the generation of TOTAL source language comments, immediately following those generated from the DESCRIPTION clauses (if applicable; otherwise, immediately following the Prologue Statements), if:

- the NOTE keyword of the DGTOT macro has a value other than 0, and the PRODUCE command does not include OMITTING NOTES; or
- the PRODUCE command includes GIVING Csl NOTES (see section 2.2.3 of the Source Language Generation manual).

If a comment line generated from a DESCRIPTION or NOTE clause is longer than the line length determined by the LENMENT keyword of the DGTOT macro, it is truncated to that length.

The Prologue Statements (with any associated comments) are followed by DBDL Master File Statements and DBDL Variable File Statements, as appropriate, generated from the data definitions of the members named in the CONTAINS clause. See section 4.4.

4.4 GENERATION FROM FILE TOTAL MEMBERS

TOTAL DBDL Master File Statements are generated from FILE TOTAL MASTER members named in the PRODUCE command, or named in the CONTAINS clause of a TOTAL-DATABASE member named in the PRODUCE command.

TOTAL DBDL Variable File Statements are generated from FILE TOTAL VARIABLE members named in the PRODUCE command, or named in the CONTAINS clause of a TOTAL-DATABASE member named in the PRODUCE command.

The name generated for a data set (tile) is the member name or alias (see section 2.2.3 of the Source Language Generation manual), subjected to any name editing specified in the PRODUCE command (see section 2.2.4 of the Source Language Generation manual), and then adjusted to four characters by removing middle characters or by appending X characters.

Data element definitions are generated for contained GROUP and/or ITEM members as determined by the presence or absence of the NOGROUPS or NO-GROUPS keyword in the PRODUCE command: see the specification of the command in section 4.2.

Data element names are generated as eight characters. The member name or alias is first subjected to any name editing specified in the PRODUCE command. Thereafter:

- if:
  - the PREFIX keyword of the DGTOT macro has the value YES and the PRODUCE command does not include OMITTING FILE-NAME-PREFIX; or
  - the PRODUCE command includes GIVING FILE-NAME-PREFIX

  then the (edited) name is adjusted to four characters by removing middle characters or by appending X characters; the resulting four characters are then prefixed with the generated name of the containing data set, to give the generated data element name.

- otherwise the (edited) name is adjusted to eight characters by removing middle characters or by appending X characters.

The length in characters of a data element is computed from the member's data definition if the member is an ITEM, or is the sum of the computed lengths of the contained items if the member is a GROUP.

If a group contained by a file specifies the keyword ALIGNED, then the binary item(s) or floating point item(s) declared (as an individual item or as elements of an array) in the content declaration is/are aligned to half word, full word or double word boundaries, thus:

- binary items having a length of four decimal digits or less occupy a complete half word
- binary items having a length of from five to nine decimal digits occupy a full word
- binary items having a length of from ten to eighteen decimal digits occupy two full words, but are not necessarily aligned to a double word boundary
- floating-point items having six digits or less in the mantissa occupy a full word
- floating-point items having from seven to sixteen digits in the mantissa occupy a double word.

UNALIGNED means that the binary item(s) or floating point item(s) declared (as an individual item or as elements of an array) in the content declaration is/are not necessarily aligned to word or half word boundaries. The amount of space occupied is the same as for ALIGNED items, but the positioning relative to word boundaries can differ. NOT-ALIGNED means the same as UNALIGNED.

Bit string items are generated in TOTAL source with byte alignment, regardless of any alignment that has been specified in the item's containing member, and regardless of the value given to the RNDBIT parameter in the DGTOT macro. Thus if any bit string items are not aligned, TOTAL source statements generated may not be consistent with record layouts that are produced from the file.

When TOTAL data element definitions are generated from GROUP members, the following rules apply:

- if the length of a member specified in an ELSE clause is longer than the member preceding the first ELSE then a filler is generated for the extra bytes

- if a member occurs more than once within the group, only one occurrence is generated, followed by a filler for the remaining occurrences

- fillers are generated for slack bytes caused by alignment specifications.

Filler names are generated with the name *FILLER*.

The control field of a master data set is generated as defined in the FILE TOTAL MASTER data definition statement specifications in section 3.3. The name generated for the control field is always the generated data set name suffixed with CTRL.

A root field definition is always generated as the first data element definition of a master data set. The name generated for the root field is always the generated data set name suffixed with ROOT.

A DEVICE statement is always generated. If the optional DEVICE clause is not present in the member's data definition, the device specified by the DEVICE keyword of the DGTOT macro is defaulted. This default is 2314 unless otherwise tailored.

If the LOGICAL-BLOCKS-PER-TRACK clause is not present in the member's data definition, an allocation of 1 is defaulted.

DESCRIPTION clauses and NOTE clauses are processed as for TOTAL-DATABASE members (see section 4.3), except that any generated comments follow the originating member's Master File or Variable File statements.

Other optional clauses in the member's data definition result in the generation of the equivalent optional TOTAL statements, provided that:

- they are valid for the version of TOTAL being produced
- they are valid (for TOTAL) in the, operating system environment (OS or DOS) under which the generation is taking place.

If a statement generated from an optional clause would be invalid for the version of TOTAL being produced, or invalid for the operating system environment, generation from that clause is suppressed.

When linkage path statements are generated from the LINKED-TO clause of a FILE TOTAL MASTER or the LINK clause of a FILE TOTAL VARIABLE, validation checks are performed to ensure that linkages are established only between master data sets and variable data sets (not between master and master, or variable and variable, data sets) and that linkage is correctly reciprocated.

When IOAREA statements are generated, checks are performed to ensure that the same IOAREA name is not produced both for a master data set and for a variable data set of the same database. These checks can be performed only when the database name is specified in the FROM clause of the PRODUCE command; they cannot be performed if only file names are specified in that clause. If the same IOAREA name is encountered in the data definitions of a master file and of a variable file for the same database, then the first one encountered is accepted; when the second one is encountered a warning message is output, and:

- for TOTAL/4, if the database has no IOAREA statement, an IOAREA name of DMIO is substituted in the file's IOAREA statement; if the database has an IOAREA statement, no IOAREA statement is output for the file
- for TOTAL/7 and TOTAL/S, no IOAREA statement is output.

4.5 GENERATION OF RECORD LAYOUTS AND PROGRAMMING SOURCE LANGUAGE FROM TOTAL-DATABASE AND FILE TOTAL MEMBERS

Record layouts, and/or programming source language data descriptions in COBOL. PL/I and Assembler, can be generated from encoded TOTAL-DATABASE. FILE TOTAL MASTER and FILE TOTAL VARIABLE members. The format of the PRODUCE command when used for these purposes is as described in the Source Language Generation manual, with the qualification that the generation control options

$$\text{GIVING} \begin{Bmatrix} \text{FD-ONLY} \\ \text{RECORDS-ONLY} \\ \text{ALL-FILE} \end{Bmatrix}$$

are not relevant when member-name is a TOTAL-DATABASE, a FILE TOTAL MASTER or FILE TOTAL VARIABLE.

When generating record-layouts and/or programming source language data descriptions from a TOTAL-DATABASE, FILE TOTAL MASTER or FILE TOTAL VARIABLE member, no record layout entries or data description statements are produced at the database or file levels: instead:

- if member-name is a TOTAL-DATABASE, each FILE named in its CONTAINS clause appears in the output as a group (at 01 level) containing the members named in the FILE's CONTAINS clause

- if member-name is a FILE TOTAL MASTER or a FILE TOTAL VARIABLE, it appears in the output as a group (at 01 level) containing the members named in its CONTAINS clause.

The tailoring macros that apply are those defined in the Source Language Generation manual (not DGTOT).

CHAPTER 5 TOTAL-DATAMANAGER CORRESPONDENCE TABLES

5.1 TOTAL DATABASE DATA DEFINITION CORRESPONDENCE TABLE

Correspondence Between TOTAL Database Data Definition Statements and **DATAMANAGER** Data **Definition Statements**

TOTAL Database Syntax	DATAMANAGER Syntax

LOGGING: {NONE
LOGSIZE=n,FILE=xxxx,
DEVICE=nnnn,
DEVADDR=SYSnnn}
	LOGGING {NONE
	FILE name SIZE I
	DEVICE device
	SYSNO n}

CTLX: {NONE YES ,DEVICE=nnnn,DEVADDR=SYSnnn}	CTLX {NONE YES DEVICE device SYSNO n}
TRACK HOLD	TRACK HOLD
SHARE IO: IOAREA=xxxx=nn	SHARE IOAREA ‘name’ b

JCL= {NO
YES}
	JCL {NO
	YES}

5.2 MASTER FILE DEFINITION CORRESPONDENCE TABLE

Correspondence Between TOTAL Master File Definition Statements and DATAMANAGER Data Definition Statements

TOTAL Syntax	DATAMANAGER Syntax
(File Prologue Statement)	
IOAREA=xxxx	IOAREA 'xxxx'
(Logical Definition Statements)	
mmmmCTRL=n	CONTROL member-name
mmmmLKxx=8	LINKED-TO file-name LINK-NAME 'xx'
xxxxxxxx=n	CONTAINS { itern-name group-name }
(Physical Definition)	
DEVICE=nnnn	DEVICE device

ACCESS-METHOD= { BDAM
VSAM }
	ACCESS-METHOD { BDAM
	VSAM }

TOTAL-LOGICAL-RECORDS=n	TOTAL-LOGICAL-RECORDS n
TOTAL-TRACKS=n	TOTAL-TRACKS n
LOGICAL-RECORD-LENGTH=n	LOGICAL-RECORD-LENGTH n
LOGICAL-RECORDS-PER-BLOCK=n	LOGICAL-RECORDS-PER-BLOCKn
LOGICAL-BLOCKS-PER-TRACK=n	LOGICAL-BLOCKS-PER-TRACK n
CONTROL-INTERVAL=n	CONTROL-INTERVAL n
CONTROL-INTERVAL-SIZE=n	CONTROL-INTERVAL-SIZE n
DISK-EXTENTS=n	DISK-EXTENTS m
OLD-FILE=YES	OLD-FILE

5.3 VARIABLE FILE DEFINITION CORRESPONDENCE TABLE

Correspondence Between TOTAL File Definition For Variable Data Sets and DATAMANAGER Data Definition Statements

TOTAL Syntax	DATAMANAGER Syntax
(File Prologue Statement)	
IOAREA=xxxx	IOAREA 'xxxx'
(Logical Definition Statements)	
mmmmLKxx=8	LINKmember-name-l LINK-NAME'xx'
xxxxxxxx=n	CONTAINS {item-name group-name}
(Physical Definition Statements)	
DEVICE=nnnn	DEVICE device

ACCESS-MEHTOD= {BDAM
VSAM}
	ACCESS-METHOD {BDAM
	VSAM}

TOTAL-LOGICAL-RECORDS=n	TOTAL-LOGICAL-RECORDS n
TOTAL-TRACKS=n	TOTAL-TRACKS n
LOGICAL-RECORD-LENGTH=n	LOGICAL-RECORD-LENGTHn
LOGICAL-RECORDS-PER-BLOCK=n	LOGICAL-RECORDS-PER-BLOCKn
LOGICAL-BLOCKS-PER-TRACK=n	LOGICAL-BLOCKS-PER-TRACK n
CONTROL-INTERVAL=n	CONTROL-INTERVAL n
CONTROL-INTERVAL-SIZE=n	CONTROL-INTERVAL-SIZE n
CYLINDER-LOAD-LIMIT=n	CYLINDER-LOAD-LIMIT p
DISK-EXTENTS=n	DISK-EXTENTS m
OLD-FILE=YES	OLD-FILE

APPENDIX 1 THE MACRO DGTOT

App.1.1 IMPLEMENTATION OF THE MACRO DGTOT

The installation macro DGTOT is available to enable TOTAL DBGEN source statements generated by the PRODUCE command to be tailored to conform to a particular installation's standards.

The macro is supplied as a source module on the installation magnetic tape. The table in App. 1.2 lists the keywords of the macro, for which values can be specified when DATAMANAGER is installed. If the supplied default values of all these keywords are acceptable, no further action need be taken in respect of the macro. If any values are to be changed, the procedure described in section 3. I of the Installation in OS Environments or Installation in DOS Environments manual must be followed. The macro assembles as the DATAMANAGER module DFU 13.

App.1.2 THE MACRO DGTOT

The following table lists the keywords of the macro DGTOT for which values can be specified when DATAMANAGER is installed.

The Macro DGTOT: Keywords Specifiable on Installation

Keyword	Specifies	Default Value	Alternative Values
ACHAR	The hexadecimal values of any additional characters that are to be accepted for output in names produced by the Source Language Generation facility, to enable characters not in the standard source language character set to be output (see Note 1)	No default	Any valid hexadecimal value. or a sub-list of such values
ACSMETH	The type of file generated by a PRODUCE command	BPAM	QSAM

(continued)

The Macro DGTOT: Keywords Specifiable on Installation

Keyword	Specifies	Default Value	Alternative Values
ATRUNK	The character part of an ALPHABETIC ITEM FILLER name	'ALPHA-FILLER'	'name' (see Note 2)
AUTOCHK	Check for and convert fillers	NO	YES
BTRUNK	The character part of a BINARY ITEM FILLER name	'BIN-FILLER'	'name' (see Note 2)
COLSEQ	Starting column of line sequence number	73	up to 99
CONCARD	Whether a control card is to be produced	YES	NO (see Note 3)
CTRUNK	The character part of a CHARACTER ITEM FILLER name	'FILLER'	'name' (see Note 2)
DDNAME	Default library-name	'GENLIB'	A delimited string of 1 to 8 characters
DESC	Maximum number of character strings of DATAMANAGER DESCRIPTION clauses to be used to generate comments	0	Up to 32767 or ALL
DEVICE	The default device type generated in the TOTAL DEVICE clause (from TOTAL file specifications)	2314	Any other valid direct access storage device type
DTRUNK	The character part of a PACKED-DECIMAL ITEM FILLER name	'DEC-FILLER'	'name' (see Note 2)
FTRLJNK	The character part of a FLOATING-POINT ITEM FILLER name	'FLOAT-FILLER'	'name' (see Note 2)
GFNL	Length of number part of group filler name	5	4 to 15
GTRUNK	Value of name part of group filler name	'GROUP-FILLER'	'name' (see Note 2)
IFNL	Length of number part of item filler name	5	4 to 14
INCLEV	Level increment	2	up to 99

(continued)

The Macro DGTOT: Keywords Specifiable on Installation

Keyword	Specifies	Default Value	Alternative Values
INCRSEQ	Line sequence number increment	10	up to 99999999
LENSEQ	Length of line sequence number	8	up to 9
LIBCC	The format of the control card output as the first record of a QSAM FILE (unless overridden by an ONTO clause in the command)	See section 2.2.2 of Source Language generation manual	A delimited character string of I to 72 characters including a question mark (?)
NOTE	Maximum number of character strings of DATAMANAGER NOTE clauses to be used to generate comments	0	Up to 32767 or ALL
NTRUNK	The character part of a NUMERIC-CHARACTER ITEM FILLER name	'NC-FILLER'	'name' (see Note 2)
MEMLEN	Maximum length of library-name	8	Up to 16
PREFIX	Whether the first four characters of generated TOTAL elements will be the generated TOTAL data-set name	YES	NO
RNDBIN	Rounding of binary items	NO	YES
RNDBIT	Whether bit string fields should be generated with byte alignment	YES	NO
SEQNO	Whether line sequence numbering required	NO	YES

Notes:

The standard Source Language Generation facility output character set for the TOTAL Interface is that defined in the TOTAL data definition language specification. This character set can be extended to allow non-standard characters to be output in names, by entering the hexadecimal value of each required character as a value to ACHAR. The user should ensure that any extra characters that are added to the output character set in this way are used only in ways that are permitted by the software with which DATAMANAGER is used.

2 name defines part of a member name. It must be stated within single quotes, must not be more than 16 characters in length, and must conform to DATAMANAGER rules for member names stated in Chapter 2 of the DATAMANAGER User's Guide. The values declared for the keywords ATRUNK, BTRUNK, CTRUNK, DTRUNK, FTRUNK, GTRUNK and NTRUNK should correspond to those of the same keywords defined in association with the Automation of Set Up facility, if that facility is also used.

3 When the value CONCARD=NO is used to suppress the generation of a control card, the production of BKEND cards is also suppressed.

USAGE DIRECTORY

ASG Worldwide Headquarters Naples Florida USA | asg.com